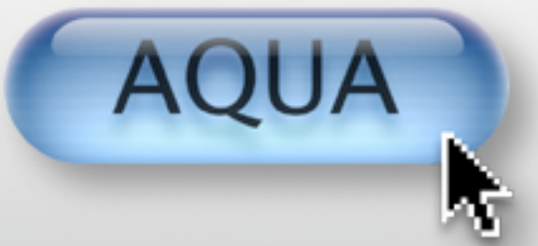
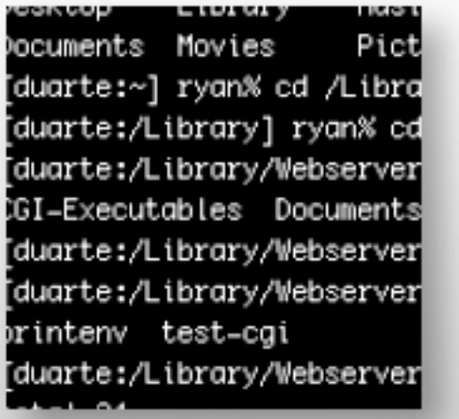
Cocoa Tips & Tricks

Session 424





Cocoa Tips & Tricks

Chuck Pisula
Cocoa Frameworks Engineer

What You Will See

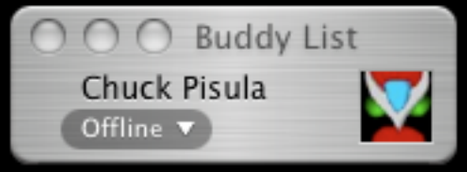
- Cool features
- Useful tips
  - From beginner to advanced
- *A lot* of code and demos
  - Full source available online

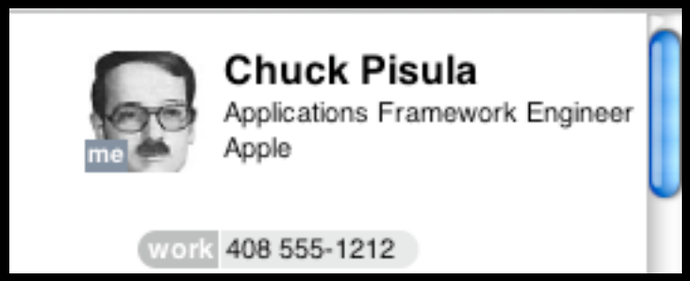
Today's Talk Format

- You ask for a tip
- We tell you the trick
- …we already thought of the questions you want to ask…

Question

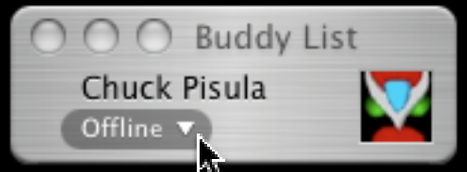
- iChat, Safari, AddressBook, and other apps have widgets with rollover effects
- How did they do that?





Question

- iChat, Safari, AddressBook, and other apps have widgets with rollover effects
- How did they do that?



Roll-overs: The Idea

- Track the mouse
  - Setup state when mouse enters/exits an area
- Custom drawing
  - Represent roll-over as bezier path

Tracking the Mouse: NSView

- Tracking all mouse move events is not necessary
- Create a tracking rect

```
- (NSTrackingRectTag)addTrackingRect:(NSRect )aRect
                               owner:(id)owner
                            userData:(void *)userData
                        assumeInside:(BOOL)flag
```

- Redo the tracking rect at appropriate times
  - Frame change, etc.
- ‘owner’ gets notified when mouse enters/exits rect

```
- (void)mouseEntered:(NSEvent *)theEvent;

- (void)mouseExited:(NSEvent *)theEvent;
```

Using NSBezierPath

- Create path with two arcs

```
- (void)appendBezierPathWithArcWithCenter:(NSPoint)center
                                   radius:(floatradius
                               startAngle:(float)startAngle
                                 endAngle:(float)endAngle
                                clockwise:(BOOL)clockwise;
```

- Close the path and

```
- (void)closePath;
- (void)fill;
```

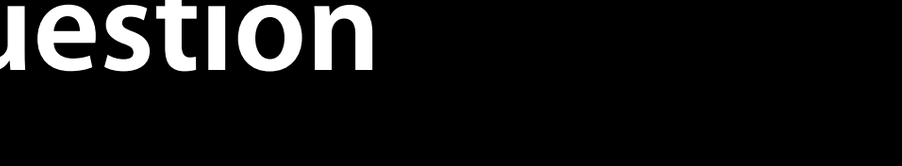
Demo

Chuck Pisula

Question

- AddressBook's editing field is raised and has shadows.
  It looks pretty cool…

  How did they do that?



Raised Editing: the Idea

- Track text selection
- Shadow border is an image
  - Draw bezier path with NSShadow style set
  - Knock out the middle
- Override -drawRect:
  - Draw text as usual with [super drawRect:]
  - Draw shadow border around selection

NSShadow—*New In Panther*

- Controlling shadow parameters

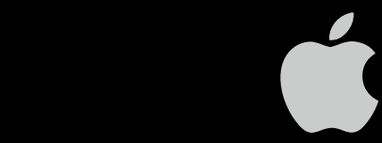
```
- (void)setShadowOffset:(NSSize)shadowOffset;

- (void)setShadowBlurRadius:(float)blurRadius;
```

- Drawing using NSShadow

- Example

```
NSShadow *shadow = ...;
[shadow setShadowOffset: ...];
[shadow setShadowBlurRadius: ...];
[NSGraphicsContext saveGraphicsState];
    [[NSColor whiteColor] set];
    [shadow set];
    [path fill];
[NSGraphicsContext restoreGraphicsState];
```



Demo

Chuck Pisula

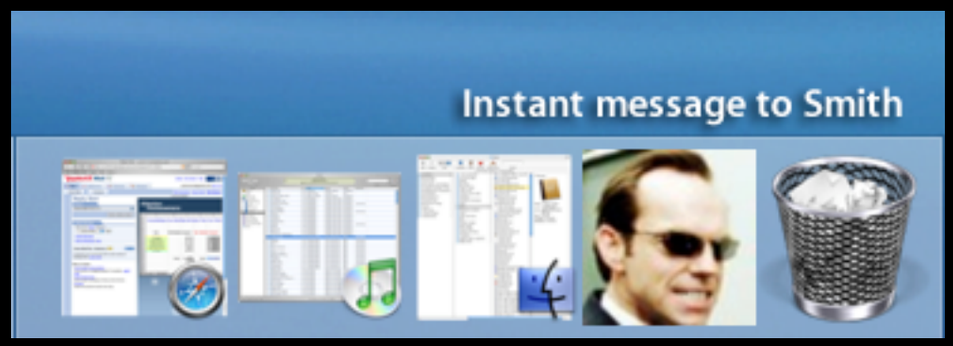
On the Hot-Seat

Tina Huang
Cocoa Frameworks Engineer

Question

- I’ve seen applications like Mail and iChat indicate application and document status in the Dock

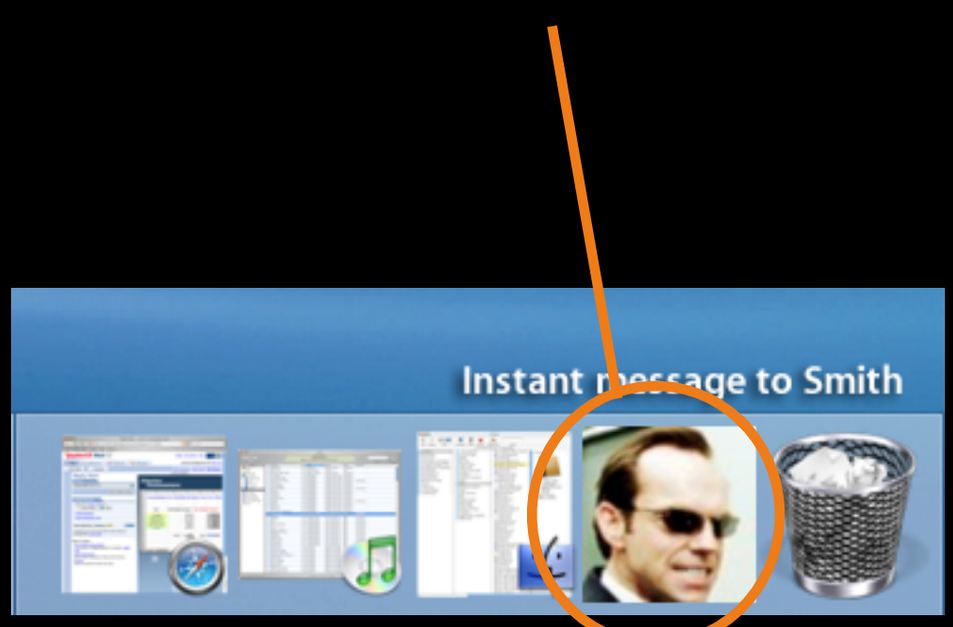
  How do they do that?



Answer

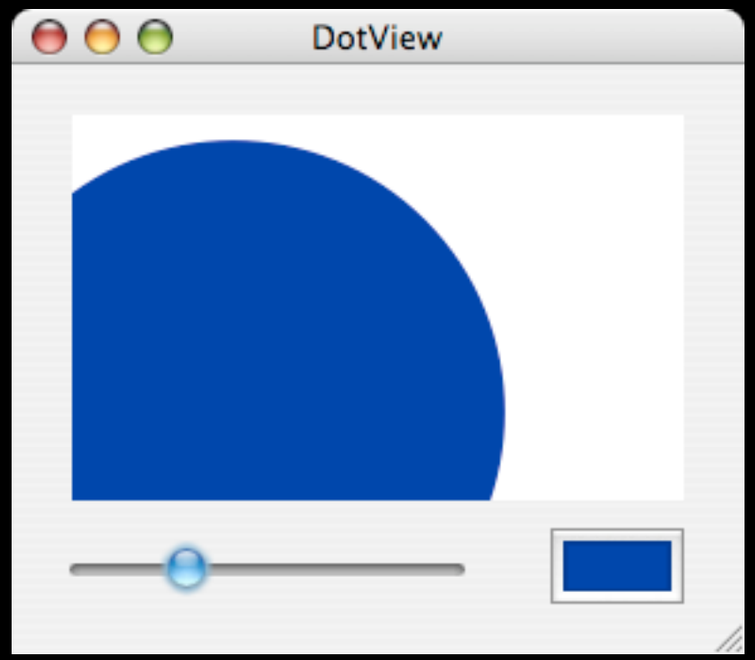
Dock Icon Badging

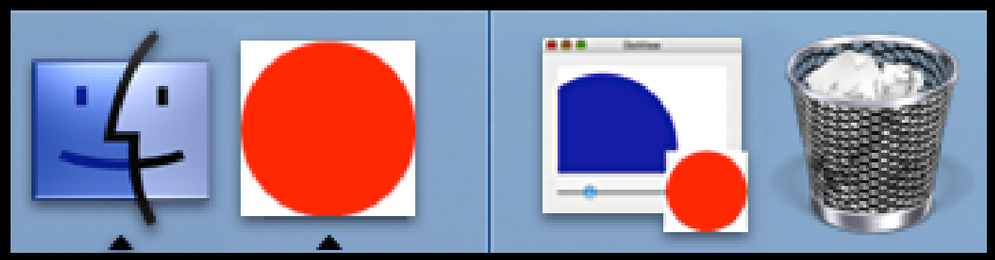
Dock Mini-Window



DotView: Best App Ever…

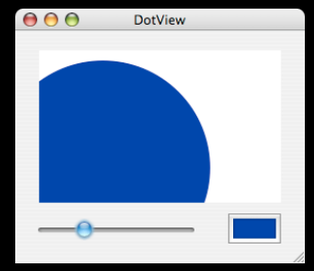
Original application



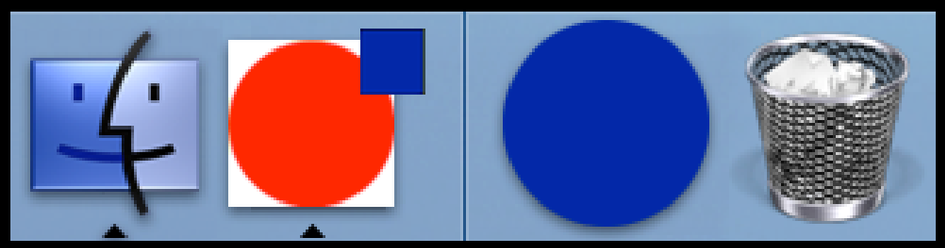


DotView: Best App Ever…

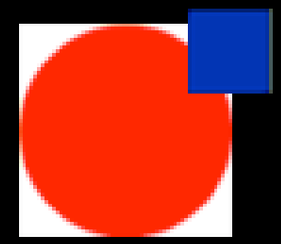
New and improved!



- Give DotView a badge to indicate the dot's color
- Make the DotView mini-window draw the dot instead of a mini-version of the window



Dock Icon Badging



- Grab the original icon

```
// NSBundle.h
+ (NSBundle *)mainBundle;
- (NSString *)bundlePath;

// NSWorkspace.h
- (NSImage *)iconForFile:(NSString *)file;
```

- Apply the badge to the icon
- Set the new icon image

```
// NSApplication.h
-(void)setApplicationIconImage:(NSImage *)newImage;
```

- Reset icon before app termination

```
// NSApplication.h
NSApplicationWillTerminateNotification
```

Demo

Tina Huang

Dock Mini-window

- Register for window notifications

```
// NSWindow.h
NSWindowDidMiniaturizeNotification

// NSView.h
- (void)viewWillMoveToWindow:(NSWindow *)window;
- (void)viewDidMoveToWindow;
```

- Create new image
- Set the image to be the mini-window image

```
// NSWindow.h
- (void)setMiniWindowImage:(NSImage *)miniMe;
```

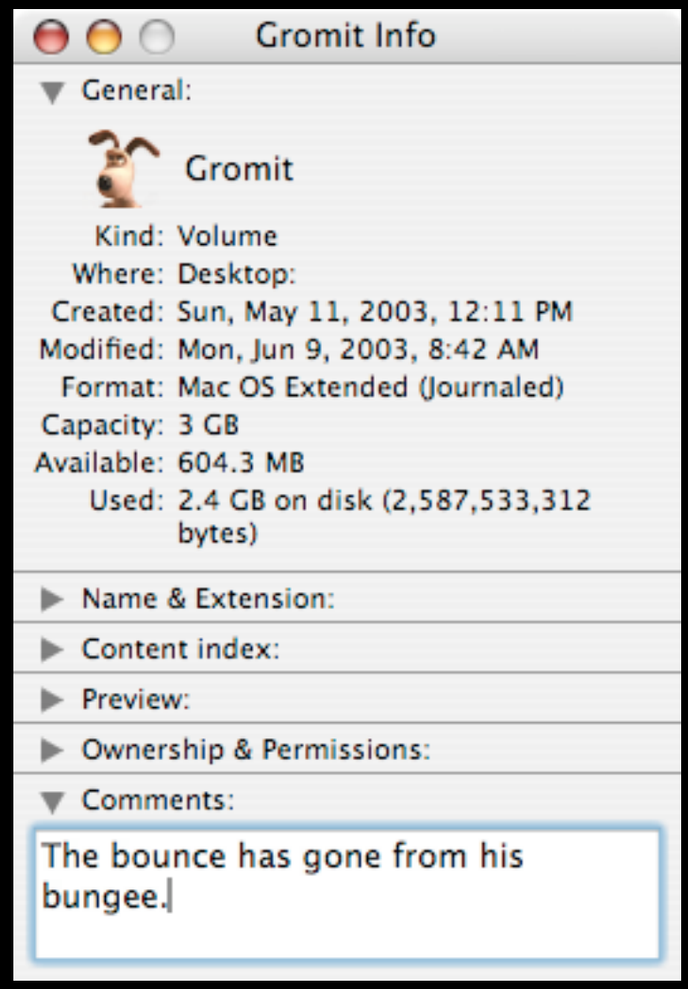
Demo

Tina Huang

Question

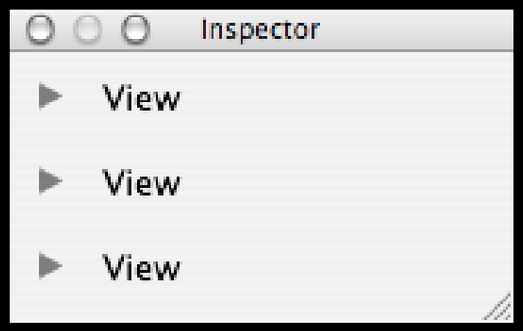
- I want my inspector to animate like the Finder's file inspector

  How can I do that?



Animating Inspector

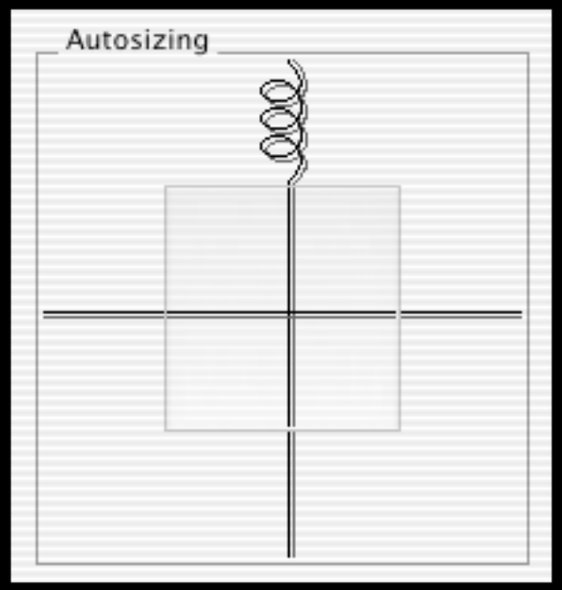
- Inspector of vertically stacked views
-
-
-



- Each view has an associated detailed view
- Temporarily adjust each view’s resizing behavior
- Resize the window with animation

Changing the Resize Behavior

Using Interface Builder

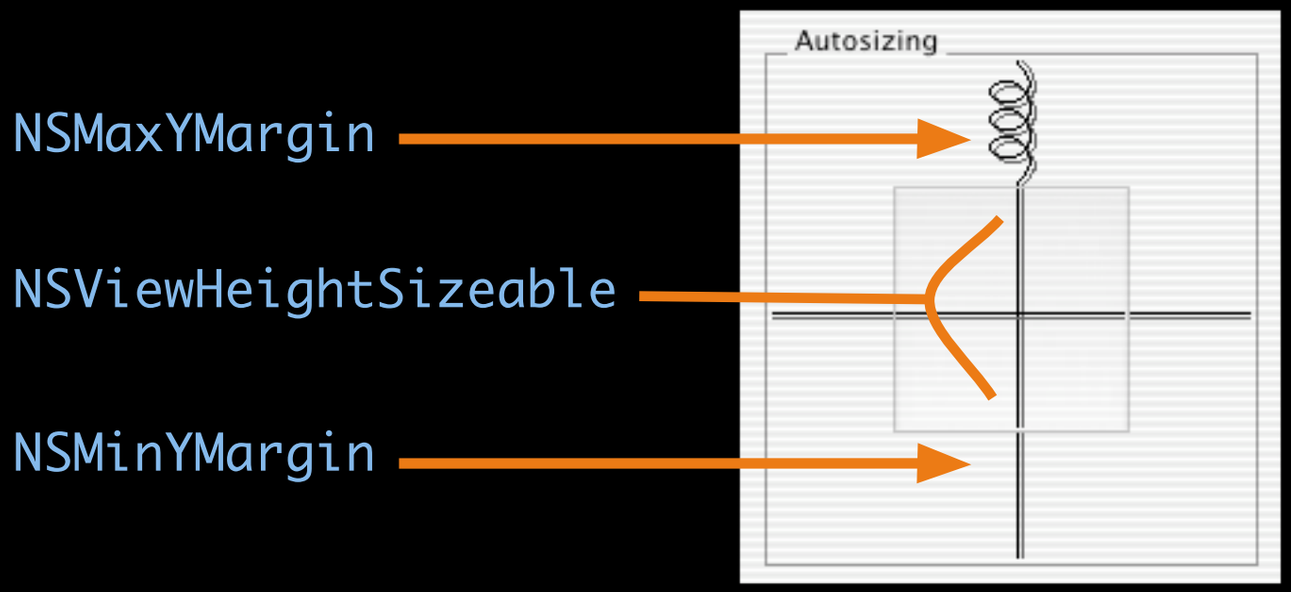


Springs specify parts that are variable

Changing the Resize Behavior

Programmatically

```
// NSView.h
- (void)setAutoresizingMask:(unsigned int)mask;
```



Mask specifies parts that are variable

Auto-resizing Masks

- Above expanding view

```
[view setAutoresizingMask:NSMinYMargin];
```

- Below expanding view

```
[view setAutoresizingMask:NSMaxYMargin];
```

- For expanding view

```
[view setAutoresizingMask:NSViewHeightSizeable];
```

- If window is resizable, only set the mask temporarily!

Demo

Tina Huang

On the Hot-Seat

Troy Stephens
Cocoa Frameworks Engineer

Full Screen OpenGL

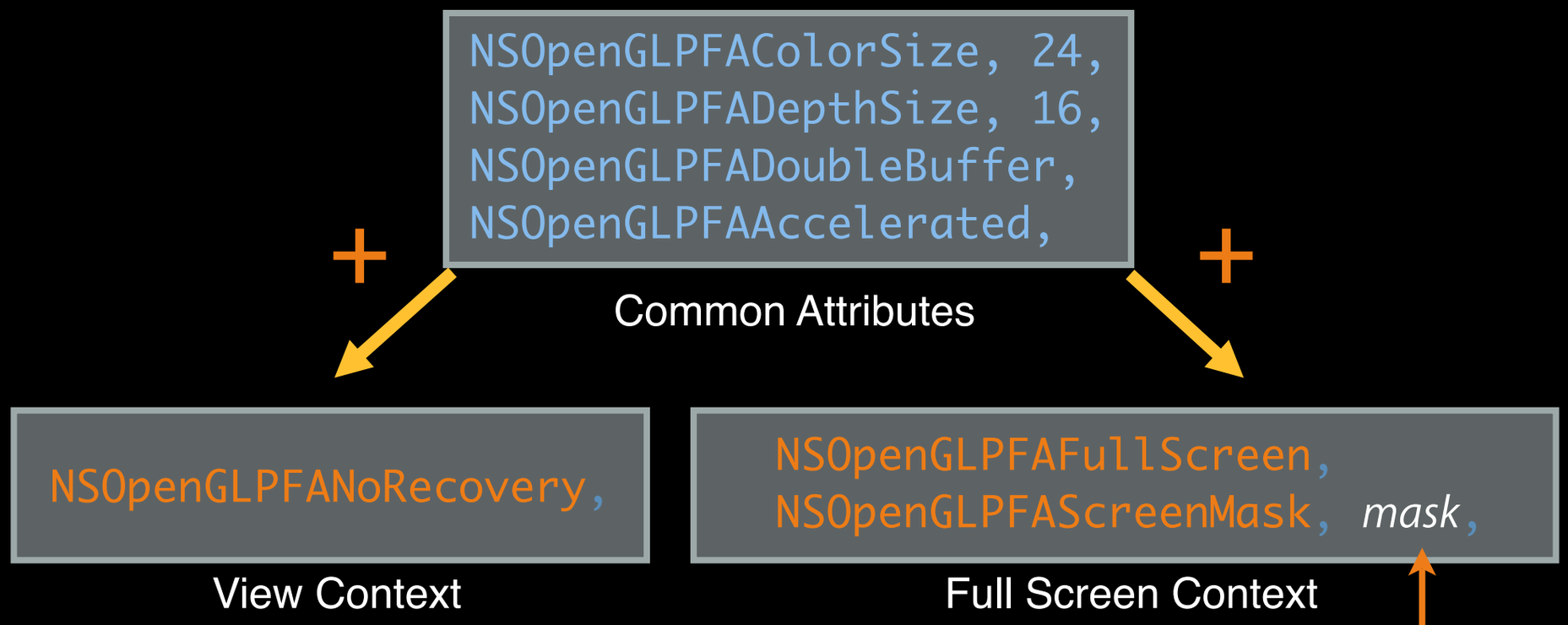
- How can my OpenGL view take over the entire screen?

Demo

Troy Stephens
tstephens@apple.com

Prerequisites

- Create a separate, full-screen `NSOpenGLContext`, with compatible pixel format attributes



Prerequisites

- Create a separate, full-screen `NSOpenGLContext`, with compatible pixel format attributes
- Drive your own loop while in full screen mode

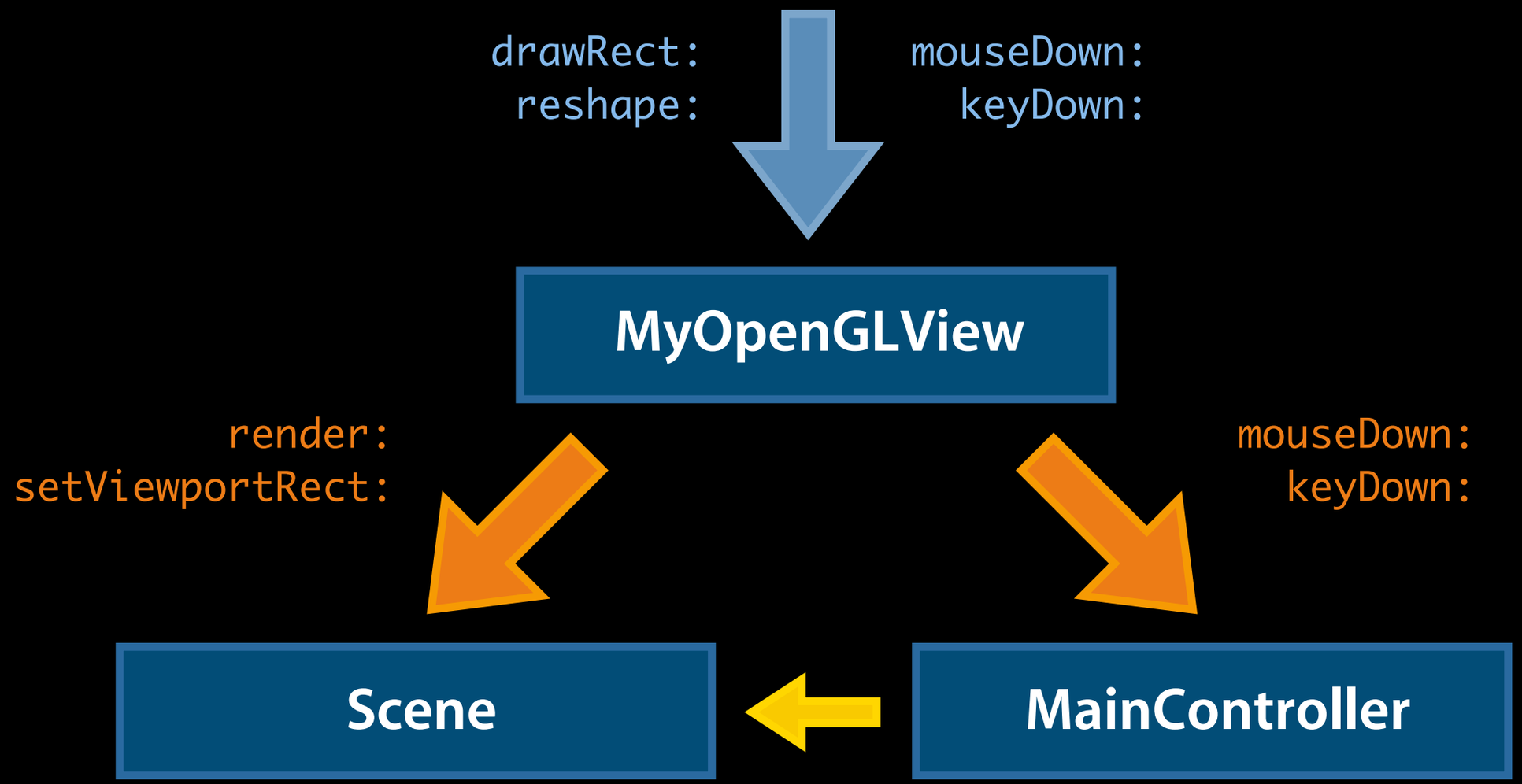
Prerequisites

- Create a separate, full-screen `NSOpenGLContext`, with compatible pixel format attributes
- Drive your own loop while in full screen mode

- Factor code out of your NSOpenGLView subclass
  - OpenGL initialization and drawing
  - input event handling

Refactoring

Move code out of your NSOpenGLView subclass



Useful API

- Capturing the display

```
CGDisplayErr CGDisplayCapture(CGDirectDisplayID display);
CGDisplayErr CGDisplayRelease(CGDirectDisplayID display);
```

- Creating an NSOpenGLContext

```
- (id) initWithFormat:(NSOpenGLPixelFormat *)format
         shareContext:(NSOpenGLContext *)share;
```

- Entering full screen mode

```
- (void) setFullScreen;
- (void) makeCurrentContext;
```

- Syncing to refresh

```
newSwapInterval = 1;
CGLSetParameter(cglContext, kCGLCPSwapInterval, &newSwapInterval);
```

Useful API

- Flushing after drawing each frame

```
- (void) flushBuffer;          // swaps frame buffers (if double-buffered)
```

- Exiting full screen mode

```
+ (void) clearCurrentContext;   // we're done rendering into this context
- (void) clearDrawable;         // exits full-screen mode
```

Source Code

Troy Stephens
tstephens@apple.com

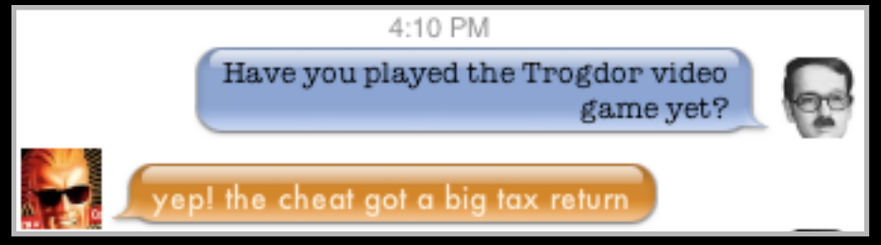
On the Hot-Seat

Doug Davidson
Cocoa Frameworks Engineer

Question

- Those iChat thought bubbles are cool!!

  How did they do that?
  How could I do something similar in my code?



Images and Text: the Idea

- Make sure the bubble gets drawn
  - Dirty rect normally just covers the text area
  - The bubble is a little bigger
- Draw the bubble before drawing the text
  - Subclass NSTextView to perform additional drawing
  - Draw the bubble after the background, before the text
- Compute the location of each bubble
  - Draw one bubble per paragraph of text
  - Need to know the area occupied by each paragraph

Drawing the Bubble

- Increase the dirty rect to cover the bubble

```
// NSTextView.h
-(NSRange)setNeedsDisplayInRect:(NSRect)rect
            avoidAdditionalLayout:(BOOL)flag;
```

- Draw the bubble before the text

```
// NSTextView.h
-(NSRange)drawViewBackgroundInRect:(NSRect)rect;
```

- Convert between character and glyph ranges

```
// NSLayoutManager.h
-(void)glyphRangeForCharacterRange:(NSRange)charRange
              actualCharacterRange:(NSRangePointer)actualCharRange;

-(void)characterRangeForGlyphRange:(NSRange)glyphRange
                actualGlyphRange:(NSRangePointer)actualGlyphRange;
```

Positioning the Bubble

- Find the range of glyphs being drawn

```
// NSLayoutManager.h
-(NSRange)glyphRangeForBoundingRect:(NSRect)r
                     inTextContainer:(NSTextContainer *)c;
```

- Find the range of characters in the paragraph

```
// NSString.h
-(void)paragraphRangeForRange:(NSRange)charRange;
```

- Union the used rectangles of each line in the paragraph

```
// NSLayoutManager.h
-(void)lineFragmentUsedRectForGlyphAtIndex:(unsigned)glyphIndex
                            effectiveRange:(NSRangePointer)effectiveGlyphRange;
```

Demo

Doug Davidson

Wrap Up

Roadmap

400 Introduction to Developing Applications with Cocoa	Nob Hill Mon., 5:00pm
410 Cocoa Update	Presidio Wed., 10:30am
412 Cocoa User Interface Programming	Presidio Wed., 3:30pm
413 Cocoa 2D Drawing Techniques	Nob Hill Wed., 5:00pm

Roadmap

419 What's New in Cocoa User Interface Programming In Depth	Russian Hill Fri., 9:00am
427 What's New in Cocoa Text	Presidio Fri., 5:00pm

Who to Contact

John Geleynse
Manager, Software Evangelism
geleynse@apple.com

Cocoa Feedback
For suggestions and feedback
cocoa-feedback@group.apple.com

Developer Technical Support
Code-level questions (fee-based)
dts@apple.com

http://developer.apple.com/wwdc2003/urls.html

For More Information

Reference Library

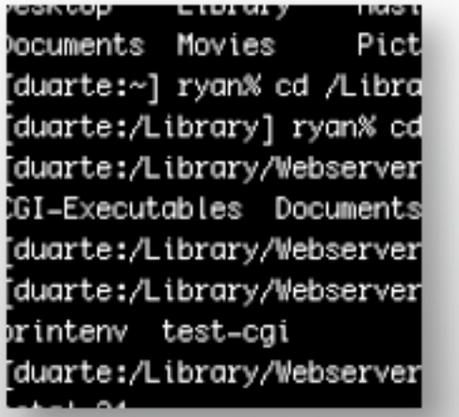
Sample Code

Demo application code *(WWDC-only distribution)*
ADC Member Site > Download Software > WWDC 2003

Documentation

Cocoa Developer Documentation
ADC Home > Documentation > Cocoa >

Q&A



http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference